

# **FTP Reference Manual**

# Table of Contents

Preface .....	4
Introduction .....	5
How to Run FTP .....	6
Basic Execution .....	6
Parallel FTP Service .....	8
Using Jumbo-Frame (High-Speed) Links .....	8
Requesting Parallel Transfers .....	10
Sample FTP Session .....	12
Standard FTP Commands .....	14
[Ap]pend .....	14
[As]cii .....	14
[Bi]nary .....	14
[By]e .....	14
Cd .....	15
[Cdu]p .....	15
[Cl]ose .....	15
[Del]ete .....	15
Dir .....	15
[Ge]t .....	16
[Ha]sh .....	16
[He]lp .....	16
Lcd .....	17
Ls .....	18
[Mde]lete .....	18
[Mge]t .....	19
[Mk]dir .....	19
[Mp]ut .....	19
[Op]en .....	20
[Par]allel .....	20
[Prom]pt .....	20
[Pu]t .....	21
[Pw]d .....	21
[Qui]t .....	22
[Quo]te .....	22
Recv .....	22
[Rem]otehelp .....	23
[Ren]ame .....	23
[Res]et .....	23
[Rmd]ir .....	23
[Rsta]tus .....	23
Send .....	24
Site .....	24
[Sta]tus .....	24

[Us]er .....	25
Common FTP Replies and Errors .....	26
FTP Reply Codes .....	30
Anonymous FTP .....	31
SFTP (Secure FTP) .....	32
Disclaimer .....	34
Keyword Index .....	35
Alphabetical List of Keywords .....	37
Date and Revisions .....	39

# Preface

**Scope:** The FTP Reference Manual describes in detail the user commands, software responses, and error codes for the FTP (File Transfer Protocol) file-transfer utility, including locally enhanced parallel file-transfer features. FTP is the standard interactive tool for moving files between machines or to (or from) the LC archival storage system (open or secure). LC's anonymous FTP service is also summarized here (open network only), as is the special encrypted-transfer tool called SFTP.

For alternative file-transfer tools that also rely on FTP software daemons to move files (but that offer special services beyond the basic FTP interface), see the [NFT Reference Manual](http://www.llnl.gov/LCdocs/nft) (URL: <http://www.llnl.gov/LCdocs/nft>) and the [HTAR Reference Manual](http://www.llnl.gov/LCdocs/htar) (URL: <http://www.llnl.gov/LCdocs/htar>) (both NFT and HTAR favor file transfers to or from storage, but they both have options that can enable more general transfers among other LC machines). For a clear comparison of FTP's features with those of NFT and SCP, and for a concise, task-oriented summary of how to use FTP for ordinary file transfers, see the [EZOUTPUT Basic Guide](http://www.llnl.gov/LCdocs/ezoutput). (URL: <http://www.llnl.gov/LCdocs/ezoutput>) For a comparison of FTP with alternative interfaces to LC's archival storage system, see the [EZSTORAGE Basic Guide](http://www.llnl.gov/LCdocs/ezstorage). (URL: <http://www.llnl.gov/LCdocs/ezstorage>)

**Availability:** FTP runs on all LC machines, open and secure. Fetch is an FTP client for Macintosh computers. Note that different vendors implement FTP details (such as listing suboptions) differently on different platforms, and system administrators can set FTP defaults (such as ascii or binary transfer mode) differently in different environments.

**Consultant:** For help contact the LC customer service and support hotline at 925-422-4531 (open e-mail: [lc-hotline@llnl.gov](mailto:lc-hotline@llnl.gov), SCF e-mail: [lc-hotline@pop.llnl.gov](mailto:lc-hotline@pop.llnl.gov)).

**Printing:** The print file for this document can be found at

OCF: <http://www.llnl.gov/LCdocs/ftp/ftp.pdf>  
SCF: [http://www.llnl.gov/LCdocs/ftp/ftp\\_scf.pdf](http://www.llnl.gov/LCdocs/ftp/ftp_scf.pdf)

# Introduction

FTP (File Transfer Protocol) is an industry-standard protocol and user interface for transferring files between computer systems by means of a series of interactive commands. FTP involves a local client (software you execute to send or receive files) and a remote server (software elsewhere that responds to instructions from your client to accept or deliver files). FTP features:

- Use of standard TCP/IP network protocols to move files between machines.
- Support for transfers to or from nonUNIX systems as well as among computers running UNIX.
- Use of IP host addresses (e.g., 134.9.50.20) as well as domain names (e.g., thunder20.llnl.gov) to specify transfer targets.
- Interactive login, usually with password, to begin transfers to or from each remote machine (at LC, some transfers are preauthenticated to omit the password).

LC users with special file-transfer needs (such as for batch-oriented command files, extensive tracking of each transfer, or persistent transfers if network problems arise) may prefer to use the NFT (URL: <http://www.llnl.gov/LCdocs/nft>) file-transfer tool to move files among LC machines. See the EZOUTPUT Basic Guide (URL: <http://www.llnl.gov/LCdocs/ezoutput>) for a more elaborate comparison of FTP and NFT. Users whose primary interest in FTP is as an interface to LC's High Performance Storage System (archival file storage) may want to consult the EZSTORAGE Basic Guide (URL: <http://www.llnl.gov/LCdocs/ezstorage>) for helpful comparisons and alternative interfaces (such as LSTORAGE, CHMODSTG, CHGRPSTG, and HTAR). Note however that starting in 2005, neither NFT nor FTP can manipulate "access control lists" (ACLs) on files stored in HPSS. If FTP transfer rates and reliability are important concerns for you (because you move large files, for example), then you can monitor recent FTP performance between many pairs of network nodes (including storage.llnl.gov) by using LC's NETMON web site (see the NETMON Reference Manual (URL: <http://www.llnl.gov/LCdocs/netmon>) for details). If you prefer a visual interface to FTP (where you select files and target directories with CTRL-CLICK using your mouse, for example), then execute HOPPER on any LC production machine and select FTP from HOPPER's CONNECT menu.

This manual tells how to run FTP, shows a typical FTP session, and describes in detail all standard FTP interactive commands (options). Standard FTP server replies and error codes are included too, along with a brief introduction to LC's anonymous FTP service. On all LC production machines (but not necessarily on LC's other machines), a parallel FTP client (PFTP) is now the default, and parallel transfers occur automatically when they are possible. Instructions for invoking a nondefault nonparallel FTP client are also included (page 6) here, along with advice on using jumbo-frame gigabit Ethernet links. Users who need to transfer files (to FIS) with their data encrypted can try SFTP, (page 32) a special FTP client with very limited server support.

# How to Run FTP

## Basic Execution

### EXECUTE LINE.

To run FTP on any LC machine, type

```
ftp [remotehost]
```

where *remotehost* is either the IP address (e.g., 134.9.50.20) or the domain name (e.g., thunder20.llnl.gov) of the computer with which you want to exchange files. The machine on which you run FTP is the "client" or "local" machine, and the machine whose address or name you specify on the execute line is the "server" or "remote" machine (for purposes of describing commands and file transfers below). If run with no *remotehost*, FTP prompts for input (and you will need to use its interactive OPEN command to specify a target host).

You must log in to your local machine to run FTP, and you must also log in to the specified remote machine at the start of each FTP file-transfer session (when you are prompted for your remote user name and password, which might be different from the local ones). FTP expects file transfers to be done by a series of interactive commands, and it does not allow "third party" transfers (between two remote machines). To adapt FTP for use in batch scripts (by means of UNIX "here files"), see the annotated batch script in the EZJOBCONTROL Basic Guide (URL: <http://www.llnl.gov/LCdocs/ezjob>). On LC production machines, HOPPER serves as a graphical controller for FTP.

### FIREWALL ALERT.

LC now uses its hardware/software security "firewall" to block direct FTP connections from machines outside the llnl.gov domain to LC machines within llnl.gov. Offsite users must either (1) log on to some llnl.gov production machine, execute FTP there, and then draw external files toward them (with the GET option), or (2) become authorized VPN users and "borrow" an llnl.gov IP address with VPN before beginning their FTP session. See EZACCESS (URL: <http://www.llnl.gov/LCdocs/ezaccess>) or the Firewall and SSH Guide (URL: <http://www.llnl.gov/LCdocs/firewall>) for detailed instructions.

### PARALLEL, NONPARALLEL CLIENTS.

On all LC production machines, open and secure (but not necessarily on LC's other machines), a parallel FTP client (PFTP) is now the default. Parallel file transfers occur automatically when they are possible. But the extra verbosity of the parallel client might sometimes pose problems (e.g., for scripts). To overtly invoke a "standard," NONparallel FTP client either type the special execute line

```
ftp.bsd [remotehost]
```

or else set the environment variable OLD\_FTP\_CLIENT to any nonnull value, for example

```
setenv OLD_FTP_CLIENT 1
export OLD_FTP_CLIENT=1
```

and then use the regular execute line shown at the start of this section (unless you include this setting in your login file, FTP will revert to the parallel default client with every new login session).

## MONITORING FTP TRAFFIC.

NETMON, LC's network-monitoring web site, tracks FTP traffic (PUTs and GETs, for example, to storage.llnl.gov) from the perspective of several benchmark OCF and SCF network nodes. NETMON tracks each HPSS Class of Service (COS) separately. In fact, NETMON's tables and graphs *always* report FTP traffic by using HPSS Class of Service categories for file size even when storage.llnl.gov is not one of the nodes generating that FTP traffic (see the NETMON COS "slang" terminology in the right-most column of the chart below; other COS values have been used in the past):

HPSS COS ID	Default for the file size	NETMON Slang term
-----	-----	-----
110	0 .GE. file .LT. 4 Mbyte	small (sFTP)
120	4 .GE. file .LT. 32 Mbyte	medium (mFTP)
135	32 .GE. file .LT. 256 Mbyte	large (lFTP)
160	256 .GE. file	jumbo (jFTP)
150	[by request only]	

The NETMON Reference Manual (URL: <http://www.llnl.gov/LCdocs/netmon>) tells how to interpret and customize NETMON's FTP-monitoring reports and plots. Starting in 2006, some Linux/CHAOS nodes are among those whose FTP traffic is monitored by NETMON.

## SECURE FTP.

For information on how executing the specialized "secure FTP" (or SFTP) client differs from running standard FTP, consult the SFTP section (page 32) below.

## TRANSFERRING ARCHIVE (TAR) FILES.

Suppose you want to bundle a set of files (perhaps including directory trees) and transfer the resulting archive to another LC machine, but you lack enough disk space to run TAR locally and (hence) double your disk usage on the client machine before you invoke FTP for the transfer. LC's special HTAR utility, formerly a storage interface only, now offers an option (-F) with which you can open a parallel connection to *any* preauthenticated LC FTP server and transfer files directly into an archive (TAR file) created and filled only on the *remote* (receiving) machine. Extractions from and even indexing of remote nonstorage archives are also supported. See the HTAR Reference Manual (URL: <http://www.llnl.gov/LCdocs/htar>) for details and annotated examples. With HTAR, archive member files can be as large as 8 Gbyte, archives can contain up to 1,000,000 member files, and there is no limit on total archive size. (HTAR manages these file transfers internally, not by executing the PFTP client.)

## CLIENT DIFFERENCES FOR HPSS ACCESS.

One common use for FTP on LC machines is to put files into or get files from archival storage (HPSS, storage.llnl.gov). But not all FTP clients interact equally well with HPSS. If you work on LC Linux/CHAOS systems (CHAOS 3.0 or later), you have access to /usr/kerberos/bin/ftp, but you should instead run /usr/bin/ftp for reaching storage. Under some conditions the former (but not the latter) client refuses to log you into HPSS or needlessly asks you to "please log in with USER and PASS."

## Parallel FTP Service

Parallel FTP service is available between each LC production machine and (both OCF and SCF) storage, and between pairs of LC production machines themselves. In all cases where parallel service is available, it is now automatic (automatic parallelization depends on the FTP daemon on the *target* machine, not on the client). Multiple "command completed" messages (one for each parallel stripe) betray the parallel transfer of large files with FTP. (HTAR also automatically uses parallel transfers but does not execute the PFTP client.)

Also, the login nodes on most LC production machines are now connected to "jumbo-frame gigabit Ethernet links" for fast network traffic. Naturally, the best file-transfer rates occur when you invoke parallel FTP between a pair of machines that also has the fast jumbo-frame links (rates between 35 and 50 Mbyte/s are possible for single FTP sessions). For example, NFT (URL: <http://www.llnl.gov/LCdocs/nft>) now automatically routes storage transfers to a cluster's login nodes to take advantage of the jumbo-frame links.

## Using Jumbo-Frame (High-Speed) Links

Network communication problems are traditionally subdivided and solved in "layers." The now-standard OSI ("open system interconnect") network model involves seven such layers, as follows:

Layer Number	Layer Name	Networking Role	Units Exchanged
7	application	supports user	
6	presentation	programs and	messages
5	session	processes	
4	transport	implements	
3	network	protocol suite	packets
2	data link	specifies network features	frames
1	physical	specifies hardware features	bits, bytes

At the lowest ("physical") layer, mere electrical signals are exchanged on the network, representing bits and bytes. But just above that layer, the network exchanges something with a meaningful internal structure, a "frame." A frame consists of your data "encapsulated" (flanked) by headers and a trailer of extra bytes used for routing, tracking, and reliability:

Ethernet header	IP header	other headers	your data	trailer	
--------------------	--------------	------------------	--------------	---------	--

A standard Ethernet frame contains 1500 bytes of user data, plus its headers and trailer. A "jumbo" frame instead contains 9000 bytes of user data, so that the percentage of overhead for the headers and trailer is much less and data-transfer rates can be much higher.



In May, 2001, LC installed "jumbo-frame gigabit Ethernet links" on both OCF and SCF storage systems, and on some production machines. Since then, jumbo frames have spread to the login nodes of most LC clusters (AIX and Linux/CHAOS). Use of these jumbo-frame network links for faster file transfers is automatic among those LC computers that have them installed. Some machines have multiple jumbo-frame links, allowing multiple high-speed file-transfer sessions at once. FTP, NFT, and HTAR clients all now check the file `/usr/local/etc/HPSS.conf` to find a machine-generated current inventory of available jumbo-frame links to use (a similar file called `/usr/local/etc/hpss_netopt.conf` supports some nonparallel legacy clients).

## Requesting Parallel Transfers

The default FTP client now on all LC production machines (but not necessarily on special-purpose machines) is a locally developed version that enables you to transfer data in parallel. When parallel FTP transfers are possible, they occur automatically. You no longer need to request parallel service or invoke a special PARALLEL command.

### AUTOMATIC.

The FTP server ("daemon") on your *destination* (target) machine determines whether or not the file transfer is automatically parallel. This is now the case for all LC production clusters (AIX or Linux/CHAOS) and storage.

For all files over 1 Mbyte, FTP file transfers to storage from all LC production machines (both directions) are automatically parallel (OCF and SCF). Transfers originating on those machines with "jumbo-frame gigabit Ethernet links" (see previous subsection (page 8)) also automatically use those links for even faster data movement. FTP's PARALLEL command (page 20) now simply reports the current parallel stripe width and block size.

### ON DEMAND.

On demand parallel transfers, once locally enabled by FTP's PARALLEL command, are no longer needed. PARALLEL now has a merely reporting role.

### PFTP COMMANDS.

The PFTP client offers nine extra commands (beyond the usual set offered by FTP) to specifically manage parallel file transfers (for example, PGET and MPGET perform parallel GETs). On LC production machines these special PFTP commands are quite unnecessary because parallel transfers occur automatically where they are possible. At other (ASC tri-lab) sites, you may need to remember the special PFTP commands to perform parallel file transfers (especially to storage). See LC's HPSS User Guide (URL: <http://www.llnl.gov/LCdocs/hpss>) for details on the nine extra PFTP commands.

### OPTIONAL EXTRAS.

LC's parallel FTP client is more verbose than the standard FTP client during file transfers. Parallel FTP users may want a complete record of each verbose FTP dialog in their batch log files. The undocumented execute-line option `-c` causes all interactive output to be sent during batch runs of FTP as well. The undocumented `-e` option copies FTP input commands into your batch output. Thus running FTP with the execute line

```
ftp -ce remotehost
```

will preserve all the details of a parallel session even within a batch job.

FTP sessions with storage (and with some other target machines) are fully preauthenticated and do not prompt for your *username*, while in other cases FTP returns a

```
Name (host:username) :
```

prompt to which you must reply to continue. Parallel users who want to eliminate this Name: prompt from all sessions (including batch sessions) can install a file called `.netrc` in their (global) home directory, containing the THREE lines

```
default  
login username
```

where last line in the `.netrc` file must be present but blank.

## Sample FTP Session

The following sample session (with annotated steps added along the right side) shows a typical dialog by which a user (JANE) transfers files interactively using FTP. In this case, the local machine (on which Jane executes the FTP client) is ATLAS, and the remote machine that files are copied to and from is LUCY. (For an alternative approach on LC production machines, you can use HOPPER as a graphical controller for FTP.)

- (1) The user runs FTP (on ATLAS) with the remote machine's domain name as an argument.
- (2) FTP prompts for a userid and a password to log in to LUCY (some LC machines "preauthenticate" and skip this password step).
- (3) At the ftp> prompt, the user changes remote directories to /var/tmp/jane (which is not shared among LC machines).
- (4) At the next ftp> prompt, the user GETs file NFT.PS (copies it from LUCY to ATLAS).
- (5) At the next ftp> prompt, the user PUTs file TESTFILE (copies it from ATLAS to LUCY).
- (6) The user then transfers a 1.1-Gbyte file called LARGE from ATLAS to LUCY. FTP automatically invokes four parallel stripes (each separately reported as FTP "completed" commands in the output).
- (7) When the file transfers are done and confirmed, the user QUITs FTP.

```
ftp lucy.llnl.gov --- (1)
```

```
Connected to lucy.llnl.gov.
220 [NOTICE TO USERS -- very long legal statement]
222 lucy.llnl.gov FTP server (Version LLNL-22...) ready.
202 Command not implemented.
Name (lucy.llnl.gov:jane): jane --- (2)
331 Password required for jane.
Password: [does not echo]
230 User jane logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
```

```
ftp> cd /var/tmp/jane --- (3)
250 CWD command successful.
```

```
ftp> get nft.ps --- (4)
200 PORT command successful.
150 Opening Binary data connection for nft.ps
226 Binary Transfer complete.
1602470 bytes received in 0.579 seconds (2.64 Mbytes/s)
```

```
ftp> put testfile --- (5)
200 PORT command successful.
150 Opening Binnary data connection for testfile
226 Transfer complete.
5264 bytes sent in 0 seconds (5.14 Kbytes/s)
```

```
ftp> put large                                     ---(6)
200 Command complete (11496780, large, 0, 4, 4194304)
200 Command complete. Address 1 is 134.9.50.35.2356
200 Command complete. Address 2 is 134.9.50.35.2357
200 Command complete. Address 3 is 134.9.50.35.2358
200 Command complete. Address 4 is 134.9.50.35.2359
150 Transfer starting.
226 Transfer complete. (moved = 11496780).
11496780 bytes sent in 0.79 seconds (16.3 Mbytes/s)
200 Command complete.

ftp> quit                                         ---(7)
221 Goodbye.
```

# Standard FTP Commands

This section lists frequently-used standard FTP commands, their definitions, usage, and examples. On many systems, you may enter the bracketed portion of the command as a shortcut. The examples use the shortcut version of the commands. Commands may only be entered at the FTP prompt, not on FTP's execute line.

Additional FTP commands might be supported by your local FTP client, and if so, they might be described in your local FTP MAN pages. Additional helpful FTP commands may also be supported by some FTP servers (such as those on LC's archival storage system). To pass an unsupported local command (such as CHMOD) to a remote FTP server that does support it, you must use FTP's QUOTE command (page 22), described below.

## [Ap]pend

Appends a local file to a file on the remote machine. If the remote file is left unspecified, the local file name is used.

Usage: `append local-file remote-file`

Example: `ap /usr/bob/newfile /users/comp/bob/file1`

## [As]cii

Sets the file transfer type to ASCII. This is commonly the default.

Usage: `ascii`

Example: `as`

Binary files should never be transferred while in ASCII mode, otherwise they will be transferred incorrectly. ASCII mode performs character translations of certain characters. These characters occur randomly in a binary file and their translation may result in the corruption of the binary file.

## [Bi]nary

Sets the file transfer type to support binary image transfer. Binary file transfer is faster than ASCII file transfer, since character and record translations are not performed. (Binary is the default for transfers to or from LC's archival storage system, open and secure.)

Usage: `binary`

Example: `bi`

## [By]e

Terminates the FTP session with the remote server and exits FTP. BYE is a synonym for QUIT.

Usage: `bye`

Example: by

## **Cd**

Changes the current directory on the remote machine to the directory specified. To change the local directory (on the client machine), use the LCD (page 17) command instead.

Usage: `cd remote-dir`

Example: `cd /users/comp/bob`

## **[Cdu]p**

Changes the current remote-machine directory to its parent directory.

Usage: `cdup`

Example: `cdu`

## **[Cl]ose**

Terminates the current FTP session with the remote server and returns to the command interpreter (the FTP prompt), but does not terminate FTP. Once you CLOSE the connection, you may use the OPEN (page 20) command to connect to another remote server.

Usage: `close`

Example: `cl`

## **[Del]ete**

Deletes the specified remote file from a remote directory. (To delete files and directories recursively, use NFT (URL: <http://www.llnl.gov/LCdocs/nft>) instead of FTP.)

Usage: `delete remote-file`

Example 1: `del file3`

Example 2: `del /users/comp/bob/dir1/file1`

## **Dir**

Lists the entries (files) and entry attributes of a directory for the remote machine. If the remote directory is not specified, the current remote directory will be listed. DIR often accepts display-control suboptions, but they vary from one FTP implementation to another. See also LS (page 18). (To list files and directories recursively, use NFT (URL: <http://www.llnl.gov/LCdocs/nft>) instead of FTP. Also, NFT's DIR with the -h option reports the class of service (COS) value for stored files in output column 3.)

Usage: `dir [remote-directory]`

Example: `dir /users/comp/bob`

## [Ge]t

Copies a remote file from a remote directory to the local machine. If no pathnames are specified, FTP will copy the *remote-file* from the current remote directory to the current local machine directory.

Usage: `get remote-file [local-file]`

Example 1: `ge /users/comp/bob/file1 /usr/bob/newfile`

Here `/users/comp/bob/file1` is copied from the remote machine to `/usr/bob/newfile` on the local machine.

Example 2: `ge file1 newfile`

Here `file1` is copied from the current remote directory to `newfile` in the current local machine directory.

Example 3: `ge file1 /usr/bob/newfile`

Here `file1` is copied from the current remote directory to `/usr/bob/newfile` on the local machine.

Example 4: `ge /users/comp/bob/file1`

Here `/users/comp/bob/file1` is copied from the remote machine to `/users/comp/bob/file1` on the local machine if the directory `/users/comp/bob` exists. If `/users/comp/bob` does not exist on the local machine, the command will return an error.

### NOTE:

If you want to GET a few files from inside a remote TAR-format archive file *without* first GETting the whole (large) archive to your local machine, use HTAR (with -F) instead of FTP. Consult the [HTAR Reference Manual](http://www.llnl.gov/LCdocs/htar) (URL: <http://www.llnl.gov/LCdocs/htar>) for instructions. You can selectively monitor FTP GET traffic by using NETMON's (URL: <http://www.llnl.gov/LCdocs/netmon>) Report Builder feature. (To GET files and directories recursively, use NFT (URL: <http://www.llnl.gov/LCdocs/nft>) instead of FTP.)

## [Ha]sh

Toggles the printing of the hash (or pound) sign (#) on the screen for each data block transferred. The size of the data block is machine dependent.

Usage: `hash`

Example: `ha`

## [He]lp

Displays information to help you learn more about FTP local client commands. There are two ways to access **help**. If you type the command **help** by itself, a list of valid FTP commands will display. To get help on a specific command, type **help** followed by the command name. Use [REMOTEHELP](#) (page 23) to get information on FTP server, rather than client, commands.



Usage 1: help

Example 1: he

Usage 2: help *command-name*

Example 2: he get

## **Lcd**

Changes the current directory on the local machine. If no directory is specified, FTP changes to your local home directory. (To change directories on the remote (server) machine, use the CD (page 15) command instead.)

Usage: lcd [*local-directory*]

Example: lcd /usr/bob

## Ls

Lists file names in a remote directory. If the remote directory is not specified, FTP lists your current remote directory. Use DIR (page 15) to list files and their attributes. To list stored files, even recursively for all levels of your storage directories, you may find the separate LSTORAGE tool more effective than LS or DIR (see EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) for instructions).

Usage: ls [*remote-directory*]

Example: ls /users/comp/bob

## [Mde]lete

Deletes multiple files. This command deletes the specified remote files. If full pathnames are not specified, the default is the current remote directory. Wildcard characters (\*, ?, [ ]) may be used with **mdelete**. To disable the interactive prompting that occurs by default for each file when you use **mdelete**, use the **prompt** command.

Usage: mdelete *remote-file1 remote-file2 ...*

Example 1: mde file1 file2

Here file1 and file2 are deleted from the current remote directory.

Example 2: mde /users/comp/bob/file1 /users/comp/barb/file1

Here /users/comp/bob/file1 and /users/comp/barb/file1 are deleted from the specified directories.

Example 3: mde test\*

Here your local FTP client expands the file filter test\* into a file list, but each remote server may process that expanded filter differently. At LC, the HPSS (storage) server for FTP interprets **mdelete** recursively, and it removes all (matching) files not only in the current remote directory (in storage) but also in the directory children of that directory as well. See the "FTP Pitfalls (with Storage)" section of EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) for several ways to handle this aggressive interpretation of **mdelete** when your remote FTP server is HPSS.

## [Mge]t

Gets multiple files. This command lets you retrieve specified files from one or more remote directories and transfer them to the current local directory. Wildcard characters (\*, ?, [ ]) may be used with **mget**. To disable the interactive prompting that occurs by default for each file when you use **mget**, use the **prompt** command.

Usage: **mget** *remote-file1 remote-file2 ...*

Example 1: **mge** file1 file2

Example 2: **mget** /users/comp/bob/file1 /users/comp/barb/file1

## [Mk]dir

Makes a directory on the remote machine. If no pathname is specified, the directory will be inserted into the current remote directory.

Usage: **mkdir** *remote-dir*

Example: **mk** /users/comp/bob/dir2

## [Mp]ut

Puts multiple files. Transfers multiple local files from one or more local directories to the current remote directory. Wildcard characters (\*, ?, [ ]) may be used with **mput**. To disable the interactive prompting that occurs by default for each file when you use **mput**, use the **prompt** command.

Usage: **mput** *local-file1 local-file2 local-file3 ...*

Example: **mp** file1 file2 file3

Here three files are transferred from the current local directory to the current remote directory.

## [Op]en

Establishes a connection to the specified host (remote) FTP server. If an optional port number is specified, FTP will attempt to contact an FTP server at that port. If you did not specify a target host when executing FTP, you must so do with the OPEN command before you can transfer files. See also CLOSE (page 15).

Usage: open host [*port*]

Example 1: op remote\_machine

Example 2: op remote\_machine 1021

## [Par]allel

Reports the current parallel stripe width and block size for automatically enabled parallel file transfers (formerly used to enable parallel transfers on LC machines, but no longer needed for that role).

Usage: parallel

Example: par

Here the FTP client responds with the current parallel block size and the current parallel stripe width. See the Parallel FTP Service (page 8) section for more details.

## [Prom]pt

Toggles interactive prompting. This occurs during multiple file transfers to allow the user to retrieve and store files selectively. If prompting is turned off, an **mget**, **mput**, or **mdelete** command operates on all files specified without prompting for your approval. The default prompt value is on.

Usage: prompt

Example: prom

## [Pu]t

Copies a local file from a local directory to the remote machine. If no pathnames are specified, PUT copies the *local-file* from the current local directory to the current remote machine directory.

Usage: `put local-file [remote-file]`

Example 1: `pu /usr/bob/file1 /users/comp/bob/newfile`

Here `/usr/bob/file1` is copied from the local machine to `/users/comp/bob/newfile` on the remote machine.

Example 2: `pu file1 newfile`

Here `file1` is copied from the current local directory to `newfile` in the current remote machine directory.

Example 3: `pu file1 /users/comp/bob/newfile`

Here `file1` is copied from the current local directory to `/users/comp/bob/newfile` on the remote machine.

Example 4: `pu /usr/bob/file1`

Here `/usr/bob/file1` is copied from the local machine to `/usr/bob/file1` on the remote machine if the directory `/usr/bob` exists on the remote machine. If `/usr/bob` does not exist on the remote machine, the command returns an error.

### NOTE:

If you want to PUT a large TAR-format archive file into storage (or to *any* LC machine with a preauthenticated FTP server) but don't have the space (or time) to build it first on your local machine, use HTAR (with -F) instead of FTP (HTAR will actually build the archive directly on the *remote* machine as member files arrive). Consult the HTAR Reference Manual (URL: <http://www.llnl.gov/LCdocs/htar>) for details. You can selectively monitor FTP PUT traffic by using NETMON's (URL: <http://www.llnl.gov/LCdocs/netmon>) Report Builder feature. (To PUT files and directories recursively, use NFT (URL: <http://www.llnl.gov/LCdocs/nft>) instead of FTP.)

## [Pw]d

Prints the name of the current remote directory.

Usage: `pwd`

Example: `pw`

## [Qui]t

Terminates the FTP session with the remote server and exits FTP. Same command as **bye**.

Usage: quit

Example: qui

## [Quo]te

Sends specified arguments or commands verbatim to the remote FTP server. Some FTP clients accept supplementary commands without a prefix (e.g., **chmod**), some clients expect **site** as a prefix for supplementary site-specific commands (such as commands to HPSS for file storage at LC), and some clients require the longer **quote site** prefix (e.g., **quote site stage**). Using the **quote site** long form is always the safest strategy to try. See [Appendix C](#) (URL: <http://www.llnl.gov/LCdocs/hpss/index.jsp?show=apc>) of the HPSS User Guide for a list of quoted commands currently accepted by the FTP server on LC's archival storage system (open and secure).

Usage: quote *arg1* [*arg2*]...

Example: quo site chmod 775 myfile

Example: [quote] site setcos 150

NOTE: if you run FTP to manage your stored files at LC, you may find it easier to use the separate tools CHMODSTG and CHGRPSTG to change permissions and storage groups, even recursively. See [EZSTORAGE](#) (URL: <http://www.llnl.gov/LCdocs/ezstorage>) for instructions. [NFT](#) (URL: <http://www.llnl.gov/LCdocs/nft>) can also change storage permissions and storage groups recursively. NFT also offers its own separate SETCOS (class of service) command better suited to use in job scripts.

## Recv

A synonym for **get**.

Usage: recv *remote-file* [*local-file*]

Example: See [GET](#) (page 16).

## [Rem]otehelp

Lists the commands supported by the FTP server to which you have connected (but not in any useful order). Commands marked with \* are unimplemented. Commands listed with quotes must be used by means of the **quote** command (but many available quoted commands may not be listed). [Rh]elp is a synonym for remotehelp. See HELP (page 16) for information client (rather than server) commands.

Usage: remotehelp [*command*]

Example: rem

Example: rem stage

## [Ren]ame

Renames the file *from-name* on the remote machine to the file *to-name*. Renaming a file is the way to move it between remote directories since no MV command is available with FTP.

Usage: rename *from-name to-name*

Example: ren /users/comp/bob/newfile /users/comp/bob/oldfile

## [Res]et

Clears the reply queue. This command resynchronizes command/reply sequencing with the remote FTP server.

Usage: reset

Example: res

## [Rmd]ir

Deletes a directory on the remote machine. If no pathname is specified, it deletes the directory from the current remote directory. You must empty a remote directory before you can delete it. (To delete directories recursively, without emptying them, use NFT (URL: <http://www.llnl.gov/LCdocs/nft>) instead of FTP.)

Usage: rmdir *remote-directory*

Example: rmd /users/comp/bob

## [Rsta]tus

Displays the current status of the FTP server (includes the version number, your user ID, the client machine name, and current data transfer mode). If a file or directory is specified, attribute information is listed as well. "PFTPD" in the response confirms a parallel FTP server (such as on LC's storage machines). See STATUS (page 24) for client information.

Usage: `rstatus` [*remote-file*]

Example: `rsta`

## Send

A synonym for **`put`** .

Usage: `send` *local-file* [*remote-file*]

Example: See PUT (page 21).

## Site

See the section above (page 22) for **`quote`** .

## [Sta]tus

Displays the current status of the FTP client (includes mode, transfer type, verbosity, case, hash settings, and the use of PORT commands). See RSTATUS (page 23) for server information.

Usage: `status`

Example: `sta`



## [Us]er

Identifies you to the remote FTP server as the same or a different user. If you enter only the login-name, you will be prompted for a password (if required). You may supply your login-name and password at the time the command is initially entered.

Usage: user *login-name* [*password*]

Example 1: user bob

User bob will be prompted to enter a password.

Example 2: us bob XXXX

User bob has entered his login-name and his password already.

# Common FTP Replies and Errors

When you enter an FTP command, you receive a corresponding reply that indicates that the command was accepted, rejected or is being processed. An FTP reply consists of a three-digit code followed by a brief description of the result. This section contains an overview of common FTP reply codes for the commands discussed in [Standard FTP Commands](#) (page 14). For a complete listing of FTP replies in numeric order, see [FTP Replies](#) (page 30).

append	<ul style="list-style-type: none"><li>• Accepted:<ul style="list-style-type: none"><li>125 Using existing data connection for <i>filename</i>.</li><li>150 Opening <i>mode</i> data connection for <i>filename</i>.</li><li>226 Transfer complete.</li></ul></li><li>• Rejected:<ul style="list-style-type: none"><li>421 Service not available, remote server has closed connection.</li></ul></li></ul>
ascii	<ul style="list-style-type: none"><li>• Accepted:<ul style="list-style-type: none"><li>200 Type set to A.</li></ul></li></ul>
binary	<ul style="list-style-type: none"><li>• Accepted:<ul style="list-style-type: none"><li>200 Type set to I.</li></ul></li></ul>
bye, close, quit	<ul style="list-style-type: none"><li>• Accepted:<ul style="list-style-type: none"><li>221 Goodbye (system status).</li></ul></li></ul>
cd, cdup	<ul style="list-style-type: none"><li>• Accepted:<ul style="list-style-type: none"><li>250 Remote CWD command successful.</li></ul></li><li>• Rejected:<ul style="list-style-type: none"><li>421 Service not available, remote server has closed connection.</li><li>550 <i>path</i>: error.</li></ul></li></ul>
delete, mdelete	<ul style="list-style-type: none"><li>• Accepted:<ul style="list-style-type: none"><li>250 Remote DELE command successful.</li></ul></li><li>• Rejected:<ul style="list-style-type: none"><li>421 Service not available, remote server has closed connection.</li><li>550 <i>filename</i>: No such file or directory.</li><li>550 <i>filename</i>: error.</li><li>550 <i>filename</i>: cannot delete .trash directory.</li></ul></li></ul>

dir	<ul style="list-style-type: none"> <li>Accepted:           <ul style="list-style-type: none"> <li>150 Opening ASCII mode data connection.</li> <li>200 Port command successful.</li> <li>226 Transfer complete.</li> </ul> </li> <li>Rejected:           <ul style="list-style-type: none"> <li>500 Data connection: <i>error</i>.</li> </ul> </li> </ul>
get, mget, recv	<ul style="list-style-type: none"> <li>Accepted:           <ul style="list-style-type: none"> <li>125 Using existing data connection for <i>remote-file</i>.</li> <li>150 Opening <i>mode</i> mode data connection for <i>remote-file (size)</i>.</li> <li>200 Port command successful.</li> <li>226 Transfer complete.</li> </ul> </li> <li>Processing:           <ul style="list-style-type: none"> <li>550 File <i>remote-file</i> is being moved from the archive.</li> </ul> </li> <li>Rejected:           <ul style="list-style-type: none"> <li>421 Service not available, remote server has closed connection.</li> <li>426 Data connection: <i>error</i>.</li> <li>550 <i>remote-file: error</i>.</li> </ul> </li> </ul>
hash	<ul style="list-style-type: none"> <li>Accepted:           <ul style="list-style-type: none"> <li>Hash mark printing on (1024 bytes/hash mark).</li> <li>Hash mark printing off.</li> </ul> </li> </ul>
help	<ul style="list-style-type: none"> <li>Accepted:           <ul style="list-style-type: none"> <li>Help text (command list).</li> </ul> </li> </ul>
lcd	<ul style="list-style-type: none"> <li>Accepted:           <ul style="list-style-type: none"> <li>Local directory now <i>pathname</i>.</li> </ul> </li> <li>Rejected:           <ul style="list-style-type: none"> <li><i>pathname: error</i>.</li> </ul> </li> </ul>
ls	<ul style="list-style-type: none"> <li>Accepted:           <ul style="list-style-type: none"> <li>150 Opening ASCII mode data connection.</li> <li>200 Port command successful.</li> <li>226 Transfer complete.</li> </ul> </li> <li>Rejected:           <ul style="list-style-type: none"> <li>426 Service not available, remote server has closed connection.</li> </ul> </li> </ul>
mkdir	<ul style="list-style-type: none"> <li>Accepted:           <ul style="list-style-type: none"> <li>257 Remote MKD command successful.</li> </ul> </li> <li>Rejected:           <ul style="list-style-type: none"> <li>550 <i>remote-directory: error</i>.</li> </ul> </li> </ul>

open	<ul style="list-style-type: none"> <li>• Accepted: <ul style="list-style-type: none"> <li>220 Remote FTP server ready.</li> <li>230 User <i>username</i> logged in, proceed.</li> </ul> </li> <li>• Rejected: <ul style="list-style-type: none"> <li>421 Service not available, remote server has closed connection.</li> </ul> </li> </ul>
prompt	<ul style="list-style-type: none"> <li>• No reply code; replies whether interactive mode is on or off.</li> </ul>
put, mput, send	<ul style="list-style-type: none"> <li>• Accepted: <ul style="list-style-type: none"> <li>125 Using existing data connection for <i>local-file</i>.</li> <li>150 Opening <i>mode</i> mode data connection for <i>local-file</i>.</li> <li>200 Port command successful.</li> <li>226 Transfer complete.</li> </ul> </li> <li>• Rejected: <ul style="list-style-type: none"> <li>421 Service not available, remote server has closed connection.</li> <li>550 <i>local-file</i>: No such file or directory.</li> </ul> </li> </ul>
pwd	<ul style="list-style-type: none"> <li>• Accepted: <ul style="list-style-type: none"> <li>257 <i>pathname</i> is current directory.</li> </ul> </li> </ul>
remotehelp, rhelp	<ul style="list-style-type: none"> <li>• Accepted: <ul style="list-style-type: none"> <li>214 Help text.</li> </ul> </li> <li>• Rejected: <ul style="list-style-type: none"> <li>502 Unknown command <i>command</i>.</li> </ul> </li> </ul>
rename	<ul style="list-style-type: none"> <li>• Accepted: <ul style="list-style-type: none"> <li>250 Remote RNTD command successful.</li> <li>350 File exists, ready for destination name.</li> </ul> </li> <li>• Rejected: <ul style="list-style-type: none"> <li>550 rename: <i>error</i>.</li> <li>550 <i>from-name</i>: No such file or directory.</li> </ul> </li> </ul>
reset	<ul style="list-style-type: none"> <li>• No reply code if accepted.</li> </ul>
rmdir	<ul style="list-style-type: none"> <li>• Accepted: <ul style="list-style-type: none"> <li>250 Remote RMD command successful.</li> </ul> </li> <li>• Rejected: <ul style="list-style-type: none"> <li>550 <i>remote-directory</i>: No such file or directory.</li> <li>550 <i>remote-directory</i>: Not a directory.</li> <li>550 Cannot delete .trash directory.</li> </ul> </li> </ul>

rstatus	<ul style="list-style-type: none"> <li>• Accepted: 221 Server status information.</li> </ul>
status	<ul style="list-style-type: none"> <li>• Accepted: Current status of FTP shown.</li> <li>• Rejected: 421 Service not available, remote server has closed connection.</li> </ul>
user	<ul style="list-style-type: none"> <li>• Accepted: 230 User logged in, proceed.</li> <li>• Processing: 331 Password required for <i>login-name</i>.</li> <li>• Rejected: 530 Login incorrect.</li> </ul>

# FTP Reply Codes

Listed below are FTP reply codes (in numeric order) and their meanings. The exact text accompanying each reply code depends on the command issued. For some examples, see [Common FTP Replies and Errors](#) (page 26), above.

Reply Code	Meaning
125	Data connection already open, transfer starting.
150	File status okay, about to open data connection.
200	Command successful.
202	Command not implemented, superfluous at this site.
213	File status.
214	Help message.
220	Service ready for new user.
221	Service closing control connection.
225	Data connection already open, no transfer in progress.
226	Closing data connection.
230	User logged in, proceed.
250	Requested action okay, completed.
257	<i>Pathname</i> is current directory. Command successful.
258	<i>Command</i> on (or off).
331	User name okay, need password.
350	Requested file action pending further information.
421	Service not available, closing control connection.
425	Can't open data connection.
426	Connection closed, transfer aborted.
451	Requested action aborted, local error in processing.
452	Requested action not taken.
500	Syntax error, command unrecognized.
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands.
504	Command not implemented for that parameter.
505	No such file or directory. File being moved from the archive.
506	Usage error.
522	Transfer error bytes written.
530	Error in user login.
550	Requested action not taken due to error.
551	Requested action aborted.
553	Requested action not taken due to system error. User not authorized to use command.

## Anonymous FTP

To support the exchange of files (especially large or nontext files) among LLNL collaborators, LC provides anonymous FTP servers on both the open and secure networks. Anyone can contact these servers (by running FTP with the user name ANONYMOUS and any password) to exchange files placed there by LC colleagues. One directory allows anyone to PUT, another allows anyone to GET, and a third directory supports PUTs and GETs exclusively for those within the llnl.gov domain. The domain name of LC's open anonymous FTP server is

`ftp.llnl.gov`

and specific instructions for using it appear in the [Anonymous FTP Service](http://www.llnl.gov/LCdocs/ezoutput/index.jsp?show=s5.4.4) (URL: <http://www.llnl.gov/LCdocs/ezoutput/index.jsp?show=s5.4.4>) section of the EZOUTPUT Basic Guide.

LC's secure-network anonymous FTP server, formerly at

`ftp.scf.cln`

has been disabled for security reasons.

# SFTP (Secure FTP)

## ROLE:

Standard FTP clients do not encrypt the data that they send to remote hosts, which theoretically allows malicious third parties to intercept and read that data. Secure FTP (SFTP) is a modified client that does encrypt all the files that it sends for greater safety. However, only suitable SFTP servers can accept file transfers from SFTP clients (because SFTP uses SSH2 and talks to the SSHD2 daemon, not to the usual FTPD or WU-FTPD daemons).

## AVAILABILITY:

- Clients--

### OCF:

SFTP clients now reside on all OCF production machines (including AIX/IBM, Linux, and Tru64/Compaq unclassified machines).

### SCF:

SFTP clients are *not* available (yet) on any LC secure-network computers.

- Servers--

### FIS:

LC's File Interchange Service (FIS, at [fis.llnl.gov](http://fis.llnl.gov)) is the only LC server that now accepts incoming files from SFTP clients. And even FIS only accepts SFTP transfers from within the LC firewall, so direct SFTP transfers from "outside" machines by means of OTS, IPA, or VPN are *not* accepted.

### Others:

No other LC FTP servers accept SFTP transfers. In particular, you can *not* store files (at [storage.llnl.gov](http://storage.llnl.gov)) from any host by running SFTP.

## DIALOG DIFFERENCES:

SFTP clients present a different user dialog than do standard FTP clients on LC machines. While some differences are trivial, others require different user responses to open connections or to transfer files successfully. SFTP...

- Does *not* request your username (nor present it as a default to which you can simply respond with RETURN).
- Checks for a "host key" for every new host to which you try to connect and, if not found, asks if you want to continue connecting (yes/no) anyway.
- Requests your one-time password (OTP) to open every no-host-key connection (no default preauthentication occurs, unlike for standard FTP connections among LC machines).
- Displays the usual long security prolog at the start of each SFTP session, but often in a distorted, unreadable format.
- Prompts for input with SFTP>.

## OPTION DIFFERENCES:

SFTP recognizes only 14 of the usual set of 35 FTP control options. (page 14) Among the most useful options that SFTP accepts are:

open, close



quit  
cd, lcd  
pwd  
get, mget  
put, mput  
mkdir, rmdir  
help [lists all SFTP options].

Among the most useful standard FTP options that SFTP does *not* accept are:

dir (to list files and their properties). In most situations, the SFTP alternative option `ls -l` reports the same information as DIR does for standard FTP sessions.

delete (to remove files). The SFTP alternatives `rm` [removes remote files], `lrm` [removes local files], perform the same functions as DELETE does during standard FTP sessions.

ascii  
binary  
parallel  
quote  
site

for which SFTP provides *no* alternative options. SFTP software is supposed to automatically detect ASCII and BINARY files on arrival and transfer them in the appropriate mode, but you cannot force the mode if inappropriate transfers occur.

#### KEYS:

If you prefer not to use your OTP (one-time password) to authenticate every SFTP session, you can create and install a special file called an "SSH public DSA key," generated using either OpenSSH or F-Secure SSH2, on every pair of machines between which you transfer files with SFTP. Generating an appropriate DSA key, converting it to the needed OpenSSH format if needed, and installing it in the right directories (including those on the open FIS node) is a complex, multi-step process. For explicit instructions on how to set up SFTP for key authentication instead of password authentication (but no other SFTP details), retrieve and print LC's Technical Bulletin 308, available on both the open and secure networks by requesting this URL with your web browser (note the 's' in https):

<https://lc.llnl.gov/computing/techbulletins/bulletin308.html>

# Disclaimer

---

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government thereof, and shall not be used for advertising or product endorsement purposes.

(C) Copyright 2007 The Regents of the University of California. All rights reserved.

---

# Keyword Index

To see an alphabetical list of keywords for this document, consult the next section (page 37).

Keyword	Description
<u>entire</u>	This entire document.
<u>title</u>	The name of this document.
<u>scope</u>	Topics covered in this document.
<u>availability</u>	Where these programs run.
<u>who</u>	Who to contact for assistance.
<u>introduction</u>	FTP features introduced.
<u>execute-line</u>	How to run FTP.
<u>ftp-usage</u>	Basic FTP client execution.
<u>parallel-usage</u>	How to request parallel FTP transfers.
<u>jumbo-frames</u>	Jumbo frames defined, inventoried.
<u>parallel-cases</u>	How parallel and jumbo FTP interact.
<u>ftp-example</u>	Typical FTP interactive session.
<u>ftp-commands</u>	FTP standard commands.
<u>append</u>	Append to a remote file.
<u>ascii</u>	ASCII (text) transfer mode.
<u>binary</u>	Binary (image) transfer mode.
<u>bye</u>	Terminate FTP.
<u>cd</u>	Change remote directories.
<u>cdup</u>	Move to remote parent directory.
<u>close</u>	Close FTP session without quitting.
<u>delete</u>	Delete remote file.
<u>dir</u>	List remote files and attributes.
<u>get</u>	Retrieve one remote file.
<u>hash</u>	Toggle hash-mark indicator.
<u>help</u>	List commands and descriptions.
<u>lcd</u>	Change local directories.
<u>ls</u>	List remote file names.
<u>mdelete</u>	Delete multiple remote files.
<u>mget</u>	Retrieve multiple remote files.
<u>mkdir</u>	Make remote directory.
<u>mput</u>	Transfer multiple local files.
<u>open</u>	Start a new FTP session.
<u>parallel</u>	Toggle parallel file transfers.
<u>prompt</u>	Toggle interactive prompting.
<u>put</u>	Transfer one local file.
<u>pwd</u>	Reveal current remote directory.
<u>quit</u>	Terminate FTP.
<u>quote</u>	Pass commands through to server.
<u>recv</u>	Retrieve one remote file.
<u>remotehelp</u>	List remote server commands.
<u>rename</u>	Rename (or move) remote file.
<u>reset</u>	Clear FTP reply queue.
<u>rmdir</u>	Delete remote directory.
<u>rstatus</u>	Show FTP server status.
<u>send</u>	Transfer one local file.
<u>site</u>	Pass commands through to server.
<u>status</u>	Show FTP client status.

<u>user</u>	Transmit user name and password.
<u>ftp-errors</u>	FTP reply and error codes explained.
<u>ftp-replies</u>	FTP reply codes by number.
<u>anonymous-ftp</u>	LC's anonymous FTP servers.
<u>sftp</u>	Secure FTP role, features, limits.
<u>index</u>	The structural index of keywords.
<u>a</u>	The alphabetical index of keywords.
<u>date</u>	The latest changes to this document.
<u>revisions</u>	The complete revision history.

# Alphabetical List of Keywords

Keyword -----	Description -----
<u>a</u>	The alphabetical index of keywords.
<u>anonymous-ftp</u>	LC's anonymous FTP servers.
<u>append</u>	Append to a remote file.
<u>ascii</u>	ASCII (text) transfer mode.
<u>availability</u>	Where these programs run.
<u>binary</u>	Binary (image) transfer mode.
<u>bye</u>	Terminate FTP.
<u>cd</u>	Change remote directories.
<u>cdup</u>	Move to remote parent directory.
<u>close</u>	Close FTP session without quitting.
<u>date</u>	The latest changes to this document.
<u>delete</u>	Delete remote file.
<u>dir</u>	List remote files and attributes.
<u>entire</u>	This entire document.
<u>execute-line</u>	How to run FTP.
<u>ftp-commands</u>	FTP standard commands.
<u>ftp-errors</u>	FTP reply and error codes explained.
<u>ftp-example</u>	Typical FTP interactive session.
<u>ftp-replies</u>	FTP reply codes by number.
<u>ftp-usage</u>	Basic FTP client execution.
<u>get</u>	Retrieve one remote file.
<u>hash</u>	Toggle hash-mark indicator.
<u>help</u>	List commands and descriptions.
<u>index</u>	The structural index of keywords.
<u>introduction</u>	FTP features introduced.
<u>jumbo-frames</u>	Jumbo frames defined, inventoried.
<u>lcd</u>	Change local directories.
<u>ls</u>	List remote file names.
<u>mdelete</u>	Delete multiple remote files.
<u>mget</u>	Retrieve multiple remote files.
<u>mkdir</u>	Make remote directory.
<u>mput</u>	Transfer multiple local files.
<u>open</u>	Start a new FTP session.
<u>parallel</u>	Toggle parallel file transfers.
<u>parallel-cases</u>	How parallel and jumbo FTP interact.
<u>parallel-usage</u>	How to request parallel FTP transfers.
<u>prompt</u>	Toggle interactive prompting.
<u>put</u>	Transfer one local file.
<u>pwd</u>	Reveal current remote directory.
<u>quit</u>	Terminate FTP.
<u>quote</u>	Pass commands through to server.
<u>recv</u>	Retrieve one remote file.
<u>remotehelp</u>	List remote server commands.
<u>rename</u>	Rename (or move) remote file.
<u>reset</u>	Clear FTP reply queue.
<u>revisions</u>	The complete revision history.
<u>rmdir</u>	Delete remote directory.
<u>rstatus</u>	Show FTP server status.
<u>scope</u>	Topics covered in this document.
<u>send</u>	Transfer one local file.
<u>sftp</u>	Secure FTP role, features, limits.
<u>site</u>	Pass commands through to server.

status  
title  
user  
who

Show FTP client status.  
The name of this document.  
Transmit user name and password.  
Who to contact for assistance.

## Date and Revisions

Revision Date -----	Keyword Affected -----	Description of Change -----
25Apr07	<u>ftp-usage</u> <u>quote</u> <u>jumbo-frames</u> <u>parallel-cases</u>  <u>parallel</u> <u>ftp-example</u>	New COS values noted. NFT's SETCOS command noted. HPSS.conf now tracks availability. Parallel transfers now automatic. Jumbo/parallel interaction simplified. Reporting role only now. Many details updated.
11Dec06	<u>ftp-usage</u>	NETMON now covers Linux/CHAOS FTP. MCR examples replaced.
10Jul06	<u>ftp-usage</u>	Warning added on client differences re HPSS.
18Jul05	<u>introduction</u> <u>ftp-usage</u> <u>ftp-example</u>	ACL support terminated. HOPPER interface noted. HOPPER interface noted.
01Sep04	<u>ftp-usage</u> <u>parallel-usage</u> <u>jumbo-frames</u>	HTAR features updated. Details updated. Linux availability added.
09Feb04	<u>introduction</u> <u>ftp-commands</u>	Use NFT to handle remote ACLs. Recursive NFT alternatives noted.
03Dec03	<u>ftp-usage</u> <u>scope</u>	HTAR size limits clarified. NFT, HTAR can do nonstorage transfers.
28Jul03	<u>ftp-usage</u> <u>get</u> <u>put</u>	HTAR now also to/from NONstorage hosts. HTAR for nonstorage archive gets too. HTAR for nonstorage archive puts too.
28Apr03	<u>parallel-usage</u> <u>parallel</u>	More automatic parallel transfers explained. Usage revised for new servers.
04Feb03	<u>sftp</u> <u>index</u> <u>introduction</u> <u>ftp-usage</u>	New section added on secure FTP. New keyword for new section. SFTP role noted. Cross reference to SFTP added.
14Nov02	<u>ftp-usage</u> <u>get</u> <u>put</u>	NETMON FTP monitoring by file size. NETMON GET monitoring available. NETMON PUT monitoring available.
09Sep02	<u>anonymous-ftp</u>	Anonymous SCF service disabled.
19Jun02	<u>mdelete</u> <u>quote</u> <u>site</u> <u>index</u>	Warning added on storage use. SETCOS example added. New section added. New keyword for new section.
04Feb02	<u>introduction</u> <u>ftp-usage</u>	NETMON cross reference added. WEST replaced in example.

	<u>ftp-example</u>	Example dialog updated.
28Aug01	<u>scope</u>	HTAR role added.
	<u>introduction</u>	HTAR role added.
	<u>get</u>	When to use HTAR instead of GET.
	<u>put</u>	When to use HTAR instead of PUT.
11Jul01	<u>parallel-cases</u>	PFTP intro, cross ref added.
26Jun01	<u>parallel-usage</u>	New section explains parallel transfers.
	<u>parallel</u>	New (LC) FTP command.
	<u>ftp-example</u>	PARALLEL command illustrated.
	<u>index</u>	New keywords for new subsections.
22Mar01	<u>introduction</u>	Default parallel clients noted.
	<u>ftp-usage</u>	How to run nonparallel client added.
	<u>ftp-example</u>	Parallel client more verbose.
	<u>status</u>	Parallel client clues noted.
	<u>rstatus</u>	Parallel server clues noted.
08Jun00	<u>scope</u>	Print instructions added.
	<u>introduction</u>	EZSTORAGE role explained.
	<u>ftp-usage</u>	Access restrictions updated.
	<u>quote</u>	CHMODSTG, CHGRPSTG alternatives noted.
	<u>ls</u>	LSTORAGE alternative noted.
	<u>anonymous-ftp</u>	New directories compared.
01Apr99	<u>execute-line</u>	Firewall blocks FTP now.
03Feb99	<u>scope</u>	SCP replaces RCP.
	<u>introduction</u>	Anonymous FTP service noted.
	<u>ftp-usage</u>	Firewall alert added.
	<u>anonymous-ftp</u>	Drop-off service deleted.
06Aug98	entire	First edition of this document.

TRG (25Apr07)



UCRL-WEB-201534

LLNL Privacy and Legal Notice (URL: <http://www.llnl.gov/disclaimer.html>)

TRG (25Apr07) Contact: [lc-hotline@llnl.gov](mailto:lc-hotline@llnl.gov)